

# Kubernetes Ingress Services by VMware NSX® Advanced Load Balancer™ (Avi Networks)

Elasticity, Security, Observability – Consolidated

## KEY BENEFITS

- **Integrated solution**  
Consolidated services including comprehensive LB, container ingress, intrinsic security, WAF, GSLB, DNS, and IPAM in one platform
- **Operational simplicity**  
A single solution with central control and ease of troubleshooting
- **Rich observability**  
Real-time telemetry with application insights, end-to-end across all components
- **Cloud-native automation**  
Elastic autoscaling based on closed-loop analytics and performance-based decision automation

## KEY FEATURES

- **Dynamic service discovery**, traffic management, and security, optimized for North-South traffic
- **Integration with Kubernetes** to automate deployment and management of container clusters
- **Multi-cluster, multi-site and multi-AZ** container cluster support across multiple geos and availability zones on a highly scalable platform

## WHAT'S INCLUDED

A single platform that provides

- Container ingress
- L4-L7 load balancing
- On-demand application scaling
- Web application firewall (WAF)
- Global server load balancing (GSLB)
- Real-time application analytics

## Kubernetes Ingress Needs a Scalable and Enterprise-Class Solution

Modern application architectures based on microservices have made appliance-based load balancing solutions obsolete. Traditional appliance-based load balancers or open source tools are not equipped to support enterprise-grade north-south ingress services, which require robust security, elastic autoscaling, integrated peripheral services and full-stack automation that are needed for a modern enterprise application architecture. This lack of a single solution results in separate products from multiple vendors to provide load balancing, ingress traffic management, DNS, IPAM and WAF services. IT faces more complex operations managing and troubleshooting multiple independent components with disparate analytics and no end-to-end visibility. These stitched together solutions necessitate in depth scripting knowledge to provide only partial automation, if any at all, leading to compromises between feature, automation, and scale.

## Challenges with Application Services for Kubernetes

Common application services, such as load balancing, network performance monitoring, and security, that are available in conventional applications often need to be implemented or approached differently in container-based applications. Here are some of such challenges in deploying container-based applications using multiple vendors.

### Multiple discrete solutions

Modern application architectures based on microservices have made appliance-based load balancing solutions obsolete. Traditional appliance-based load balancers or open source tools are not equipped to support the north-south interactions between the services, do not support application autoscaling, and lack the native integration with peripheral services, such as DNS / IPAM and WAF.

### Complex operations

With multi-vendor solutions IT faces more complex operations in managing and troubleshooting multiple independent components from different vendors.

### Lack of observability

End-to-end visibility is especially important with container-based applications. Application developers and operations teams alike need to be able to view the interactions between the peripheral services and the container services to identify erroneous interactions, security violations, and potential latencies.

### Partial automation

Application and networking services need to be API-driven and programmable without the constraints of hardware appliances or multi-vendor solutions that limit their flexibility and portability across environments. Multi-vendor solutions also necessitate in depth scripting knowledge for different products to provide only partial automation, if any at all, leading to compromising between feature, automation, and scale. Therefore, it's necessary to have consolidated Kubernetes services from a single platform (see Figure 1).

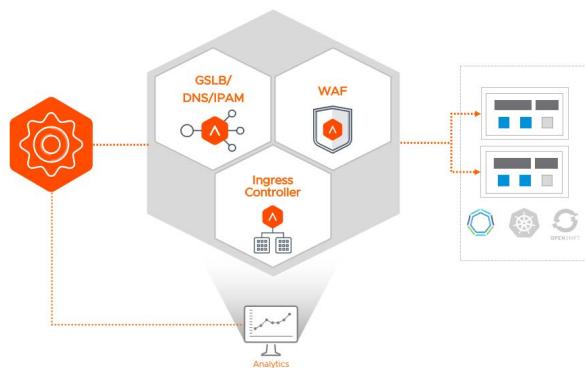


FIGURE 1: Avi Consolidated Kubernetes Services

### Consolidated Services Fabric for Container Ingress

VMware NSX Advanced Load Balancer integrates with container orchestration platforms such as Kubernetes/OpenShift/Tanzu on virtual machines and bare metal servers across on-prem, multi-cloud, multi-cluster, and multi-region environments. To deliver comprehensive container services for both traditional and cloud-native applications, Avi Kubernetes Services is optimized for North-South (ingress controller) traffic management, performance monitoring, dynamic service discovery and security including local and global server load balancing (GSLB), web application firewall (WAF), and DNS/IPAM management. Combining L4 through L7 load balancing, GSLB, DNS/IPAM management, and security functionalities in a single solution, Kubernetes Ingress Services provides operational consistency regardless of which on-prem, private-cloud or public-cloud environment the Kubernetes cluster is running on (see Figure 2).

Kubernetes Ingress Services is based on a software-defined, distributed architecture with four major components:



**Avi Controller:** The Avi Controller is the central management component of the Avi architecture providing all control plane functionality of infrastructure orchestration, centralized management, and the analytics dashboard. In Kubernetes environments, the Avi Controller is in lock steps with Kubernetes master in a scalable manner. It can be deployed anywhere as long as connectivity and latency requirements are satisfied.



**Avi Service Engines:** In Kubernetes environments, the SEs are deployed external to the cluster and provide services such as LB, GSLB, analytics, DNS and WAF in the data plane.



**Avi Kubernetes Operator:** AKO is a pod running in Kubernetes clusters that provides communications with Kubernetes master to provide configuration. AKO remains in sync with the required Kubernetes objects and calls the Avi Controller APIs to deploy the Ingress Services via the Avi Service Engines.



**Avi Multi-Kubernetes Operator:** The AMKO facilitates multi-cluster application deployment extending application ingress controllers across multi-region and multi Availability Zone deployments. AMKO calls Avi APIs for Avi Controller to create GSLB services on the leader cluster which synchronizes with all follower clusters.

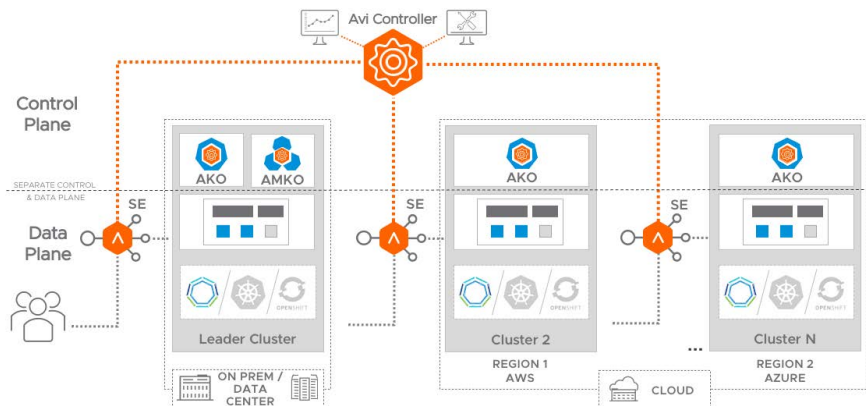


FIGURE 2: Application Services Architecture for Containers

### Kubernetes/OpenShift/Tanzu Cluster Integration using Avi Kubernetes Operator

Avi Kubernetes Ingress Services can be used for integration with multiple Kubernetes clusters, with each cluster running its own instance of AKO. AKO is a pod running in Kubernetes clusters that provides communications with Kubernetes master to provide configuration. To extend applications across Multi Region and Multi Availability Zone deployments AMKO is required.

AKO synchronizes required Kubernetes objects and calls the Avi Controller APIs to deploy and configure the Ingress Services via the Avi Service Engines (see Figure 3). Clusters are separated on SEs, which are deployed outside the cluster in the Data Plane by using VRF Contexts. Automated IPAM and DNS functionality is handled by the Avi Controller.

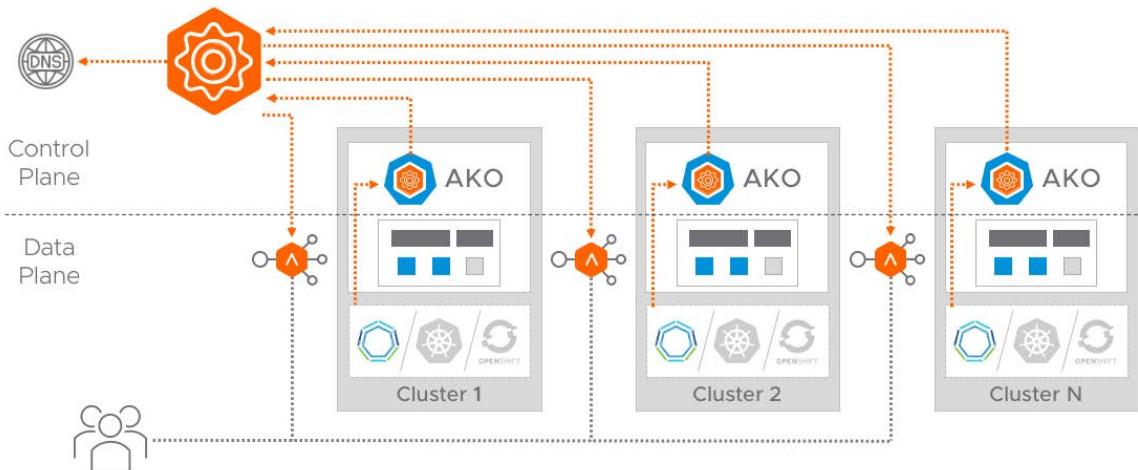


FIGURE 3: AKO Service Architecture

### Kubernetes/OpenShift/Tanzu GSLB Integration using Avi Multi-Cluster Kubernetes Operator

AMKO is an Avi pod running in the Kubernetes/OpenShift/Tanzu GSLB leader cluster and in conjunction with AKO, AMKO facilitates multi-cluster application deployment, mapping the same application deployed on multiple clusters to a single GSLB service, extending application ingresses across Multi Region and Multi Availability Zone deployments.

AMKO calls the Avi Controller APIs to deploy and configure GSLB services and DNS/IPAM settings which tie together the virtual services created by AKO on the leader and follower Kubernetes/OpenShift/Tanzu cluster sites (see Figure 4).

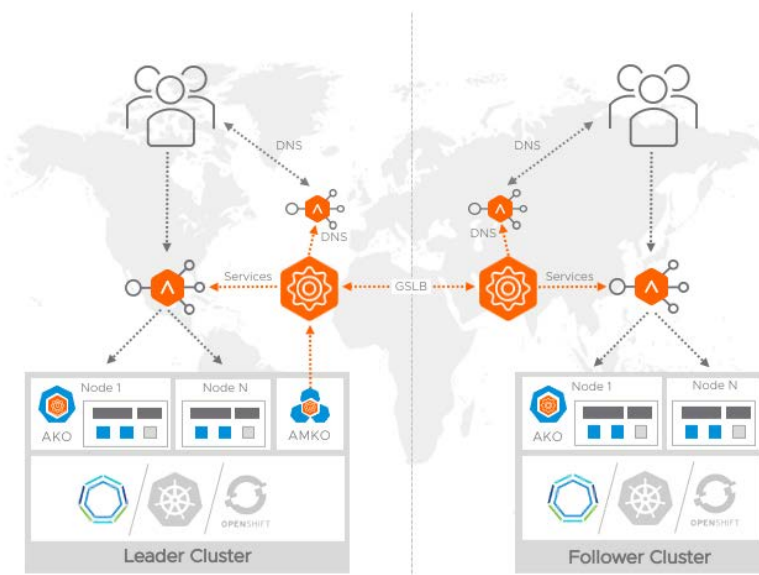


FIGURE 4: AMKO Service Architecture

## Features at a Glance



### Predictive Autoscaling

- Autoscaling of load balancers and apps based on traffic patterns
- Ability to deploy a load balancer with different capacity in real time
- Automated service discovery
- Dynamically manages IPAM/DNS for discovered newly created/deleted/updated ingress controllers



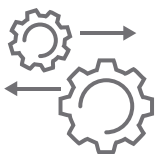
### Universality

- Centrally orchestrated services with load balancing, service discovery, security, and analytics
- Multi-Infra: Traditional and cloud-native apps in VMs/bare metal/containers
- Multi-Cluster: Inter/intra container cluster management and secure gateways
- Multi-Region: GSLB for multiple regions and geo-aware load balancing
- Multi-Cloud: Across on-premises data centers and multi-region public cloud



### Security

- Distributed Web Application Firewall (WAF) for application security
- Single sign-on (SSO) integration for enterprise-grade authentication and authorization
- Positive security model and application learning for automated acceptlist/denylist policies
- DDoS detection and mitigation for Layer4 and Layer7 attacks
- Transaction tracing & logging



### Traffic Management

- Advanced Kubernetes ingress controller with integrated IPAM/DNS
- L4-7 load balancing with SSL/TLS offload
- North-south traffic management with content switching, redirection, caching, and compression
- CI/CD and application upgrades using Blue-Green or canary deployment



### Observability

- Machine learning-based insights and app health analytics
- Application and infra performance metrics
- Health Monitoring of cluster connectivity and performance
- Real-time application and container performance monitoring with tracing